Chapter 4

Statistics

In this course, we will be examining large data sets coming from different sources. There are two basic types of data:

- Deterministic Data: Data coming from a specific function- each point has a unique image. If we know the function, we can predict exactly the data terms that we see.

- Probabilistic Data: This data cannot be predicted exactly; with similar inputs, we might find very different outputs. The data is *random* in some sense. In this case, we can only hope to model some characteristics of the data, and not the exact data points themselves. We saw this earlier in the $N-$armed bandit problem, where we saw, with each pull of the arm, we would get different outputs. Thus, we were interested in extracting general information about the processes- the expected values, or mean payoff of each machine.

In data analysis, most data falls in between the two classes- that is, data from most sources involves a deterministic part, and a random part (perhaps from measurement errors).

We will not do a comprehensive listing of statistical procedures here. Instead, we introduce this material so that we have some statistical language to work with later on.

We enter this section with some words of caution: Our notation may be different than the notation you'll see in statistics books. For example, we should use different notation for the *sample* mean and variance versus the *population* mean and variance. However, we'll use the measures sparingly and we won't be getting too far into the statistics; therefore, we'll feel free to be informal with our notation.

4.1 Functions that Define Data

The basic way of defining non-deterministic data is through the use of a *probability density function*, or p.d.f. Rather than defining a function in terms of inputs and outputs, a p.d.f. defines the probability of certain events occurring.

To be more specific, a function $f(x)$ is said to be a probability density function if it satisfies the following conditions:

1. $f(x)$ is always non-negative.
2. $\int_{-\infty}^{\infty} f(x)\,dx = 1$
3. The probability of an event between values $x = a$ and $x = b$ is given by:

$$\Pr(a \leq X \leq b) = \int_a^b f(x)\,dx$$

From the definition, we see that the probability of any specific number is zero. Furthermore, in practice we won't be dealing with continuous functions (although we might try modeling them); rather, we will be looking at discrete intervals.

Therefore, suppose that our function f is non-zero on a finite interval (another way to say this is that f has bounded support). Then we can break the interval (or intervals) up into subintervals, and consider the probability of data occurring in each of these. In this case, the p.d.f. is discrete, and our definition changes somewhat:

A discrete p.d.f. will be a finite set of numbers, $\{P_1, P_2, \ldots, P_k\}$, so that:

1. P_i is non-negative, for all i.
2. $\sum_{i=1}^{k} P_k = 1$
3. The probability of a data value occurring in subinterval i (or bin i) is P_i.

Note that in this case, each of our "events" (data in subinterval i) are disjoint, as the probability of landing on an endpoint is zero. In the $N-$armed bandit problem, we had some experience with these- in that case, $P_i = \pi(i)$, which was the probability of choosing machine i.

In Matlab, we visualize a PDF via the `hist` command, which produces a histogram of the input. Let's take a look at some template probability distributions:

1. **Example 1: The Uniform Distribution**
   - The Continuous Version:

$$f(x) = \begin{cases} \frac{1}{b-a}, & \text{if } a \leq x \leq b \\ 0, & \text{otherwise} \end{cases}$$

- The Discrete Version (using N bins over the same interval):

$$\Pr\left(a+(i-1)\frac{b-a}{N} \le x \le a+i\frac{b-a}{N}\right) = \frac{1}{N} = P_i, i = 1, 2, \ldots, N.$$

- In Matlab, to obtain a value from a uniform distribution over $[0, 1]$, we type $x =$ `rand`

2. **Example 2: The Normal (or Gaussian) Distribution**

   - The Continuous Version: The Normal distribution with mean μ and variance σ^2 (to be defined shortly) is defined as:

$$f(x) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(\frac{-(x-\mu)^2}{\sigma^2}\right)$$

     This is the common "bell-shaped curve"; the constant in the front is needed to make the integral evaluate to 1. Note that the normal distribution with zero mean and variance 1 simplifies to:

$$f(x) = \mathcal{N}(0,1) = \frac{1}{\sqrt{2\pi}}\,\mathrm{e}^{-x^2}$$

   - In Matlab, we can obtain values from a normal distribution with zero mean and unit variance by $x =$ `randn`.

3. Our last example we define only in the continuous case. It is the p.d.f. commonly used to model the human voice, and is called the double Laplacian distribution:

$$f(x) = \begin{cases} K\mathrm{e}^{-|x|}, & -1 \le x \le 1 \\ 0, & \text{otherwise} \end{cases}$$

   In the exercises, you'll be asked to determine the value of K, and we'll also see how the shape of the Laplacian compares to a normal distribution.

4.1.1 The probability distribution function

The probability distribution function (a.k.a. distribution function, cumulative distribution function) is defined via the probability density function:

$$F(X) = \Pr(-\infty < X < x) = \int_{-\infty}^{x} f(t)\,dt$$

We note that:

- By the Fundamental Theorem of Calculus, part I, $F(x)$ is the antiderivative of $f(x)$.
- $F(x)$ is strictly increasing, going from a minimum of zero to a maximum of 1.

- We saw the discrete version of this when we were working with the $N-$armed bandit; we used the Matlab function `cumsum` to create the cumulative distribution.
- To minimize confusion between terms, we'll refer to $F(x)$ as the distribution function, or cumulative distribution function.

4.2 The Mean, Median, and Mode

The most basic way to characterize a data set is through one number- the mean (or median or mode). Let's define these terms, using the data $\{x_1, x_2, \ldots, x_m\}$:

- The *Sample Mean* is:

$$\mu = \frac{1}{m}\sum_{k=1}^{m} x_k$$

We're all familiar with this definition- it's just the average of the data points. We will call this the mean, even though there are some issues here- The definition of the *population mean* requires knowledge of the underlying p.d.f., which we seldom have. In that case, the definition (a.k.a. the Expected value) is given by:

$$\mu = E(X) = \sum_{\text{all x}} xf(x)$$

You might remember this formula by seeing that this is really a weighted average, with those events most probable getting a higher weight than events less probable.

Also, if your data is being drawn independently from a fixed p.d.f., then the sample mean will converge to the population mean, as the number of samples gets very large.

Suppose we have m vectors in $\mathbb{R}^n$. we can similarly define the mean, just replace the scalar x_k with the k^{th} vector:

$$\boldsymbol{\mu} = \frac{1}{m}\sum_{k=1}^{m} \boldsymbol{x}^{(k)}$$

In Matlab, the mean is a built-in function. The command is `mean`, and the output depends on whether you input a vector or a matrix of data.

For vectors, `mean(x)` outputs a scalar.

```
m=mean(X,1);  %Returns a row vector
m=mean(X,2);  %Returns a column vector
```

See the end of this section for more on mean subtracting a matrix of data.

- The *Median* is a number so that exactly half the data is above that number, and half the data is below that number. Although the median does not have to be unique, we follow the definitions below if we are given a finite sample:

  If there are an odd number of data points, the median is the middle point. If there is an even number of data points, then there are two numbers in the middle- the median is the average of these.

  Although not used terribly often, Matlab will perform the median as well as the mean:

  ```
  m=median(X);
  ```

  where the output is a scalar if X is a vector, or a row vector if X is a matrix.

- The *Mode* is the value taken the most number of times. In the case of ties, the data is multi-modal.

We'll compare these definitions in the Exercises.

4.2.1 Centering and Double Centering Data

Let matrix A be $n \times m$, which may be considered n points in R^m or m points in $\mathbb{R}^n$. If we wish to look at A both ways, a double-centering may be appropriate.

The result of the double-centering will be that (in Matlab), we determine $\hat{A}$ so that

$$\texttt{mean}(\hat{A}, 1) = 0, \qquad \texttt{mean}(\hat{A}, 2) = 0$$

The algorithm is (in Matlab):

```
%Let A be n times m
[n,m]=size(A);
rowmean=mean(A);
A1=A-repmat(rowm,n,1);
colmean=mean(A1,2);
Ahat=A1-repmat(colmean,1,m);
```

or, equivalently:

```
%Let A be n times m
[n,m]=size(A);
colmean=mean(A,2);
A1=A-repmat(colmean,1,m);
rowmean=mean(A1,1);
Ahat=A1-repmat(rowmean,n,1);
```

Proof: For the first version (row mean first):

Let A_1 be the matrix A with the row mean **b** subtracted:

$$A_1 = \begin{bmatrix} a_{11} - b_1 & a_{12} - b_2 & \cdots & a_{1m} - b_m \\ a_{21} - b_1 & a_{22} - b_2 & \cdots & a_{2m} - b_m \\ \vdots & & & \vdots \\ a_{n1} - b_1 & a_{n2} - b_2 & \cdots & a_{nm} - b_m \end{bmatrix}$$

with

$$b_j = \frac{1}{n} \sum_{i=1}^{n} a_{ij}$$

Now define **c** as the column mean of A_1. Mean subtraction of this column results in the $\hat{A}$, written explicitly as:

$$\hat{A} = \begin{bmatrix} a_{11} - b_1 - c_1 & a_{12} - b_2 - c_1 & \cdots & a_{1m} - b_m - c_1 \\ a_{21} - b_1 - c_2 & a_{22} - b_2 - c_2 & \cdots & a_{2m} - b_m - c_2 \\ \vdots & & & \vdots \\ a_{n1} - b_1 - c_n & a_{n2} - b_2 - c_n & \cdots & a_{nm} - b_m - c_n \end{bmatrix}$$

By definition, the column mean of $\hat{A}$ is zero. Is the new row mean zero? It is clear that the new row mean is zero iff $\sum_k c_k = 0$, which we now show:

Proof that $\sum_{k=1}^{n} c_k = 0$

We explicitly write down what c_k is:

$$c_k = \frac{1}{m} \sum_{j=1}^{m} (a_{kj} - b_j)$$

and substitute the expression for b_j,

$$c_k = \frac{1}{m} \sum_{j=1}^{m} \left(a_{kj} - \frac{1}{n} \sum_{i=1}^{n} a_{ij} \right) = \frac{1}{m} \sum_{j=1}^{m} a_{kj} - \frac{1}{mn} \sum_{j=1}^{m} \sum_{i=1}^{n} a_{ij}$$

Now sum over k:

$$\sum_{k=1}^{n} c_k = \sum_{k=1}^{n} \left(\frac{1}{m} \sum_{j=1}^{m} a_{kj} - \frac{1}{mn} \sum_{j=1}^{m} \sum_{i=1}^{n} a_{ij} \right) =$$

$$\frac{1}{m} \sum_{k=1}^{n} \sum_{j=1}^{m} a_{kj} - \frac{n}{mn} \sum_{j=1}^{m} \sum_{i=1}^{n} a_{ij} = 0$$

It may be clear that these two methods produce the same result (e.g., row subtract first, then column subtract or vice-versa). If we examine the (i, j)th entry of $\hat{A}$,

$$\hat{A}_{ij} = a_{ij} - b_j - c_i = a_{ij} - \frac{1}{n} \sum_{k=1}^{n} a_{kj} - \frac{1}{m} \sum_{k=1}^{m} a_{ik} + \sum_{r=1}^{m} \sum_{s=1}^{n} a_{rs}$$

Therefore, to double center a matrix of data, each element has subtracted from it its corresponding row mean and column mean, and we add back the average of all the elements.

As a final note, this technique is only suitable if it is reasonable that the $m \times n$ matrix may be data in either $\mathbb{R}^n$ or $\mathbb{R}^m$. For example, you probably would not double center a data matrix that is 5000×2 unless there is a specific reason to do so.

Exercise: Experimentally verify the results of this section by performing (in Matlab) three ways of double-centering the data on a random 6×8 matrix A (it should not already have mean zero in either columns or rows- you should check that first).

- Subtract the row mean of A, then compute and subtract the column mean.
- Subtract the column mean of A, then compute and subtract the row mean.
- Compute the row mean and column mean and overall mean of the matrix A. Subtract row means, then column means, then add in the overall mean.

4.3 The Variance and Standard Deviation

The number that is used to describe the spread of the data about its mean is the *variance*:

Let $\{x_1, \ldots, x_m\}$ be as defined above. Then the *Sample Variance* is:

$$\sigma^2 = \frac{1}{m} \sum_{k=1}^{m} (x_k - \mu)^2$$

where μ is the mean of the data. If we think of the data as having zero mean, and placing each data point in a vector of length m, then this formula becomes:

$$\sigma^2 = \frac{1}{m} \|\boldsymbol{x}\|^2$$

NOTE: Some writers define the variance by using $\frac{1}{m-1}$ rather than $\frac{1}{m}$ (for example, see the discussion in [**?**]). We use the definition above, as it will be consistent with our other uses of the variance.

While we're defining the terms, we should distinguish between the sample variance and the actual population variance (as we did in the mean). The population variance is defined as the expected value of $(x - \mu)^2$,

$$E((x - \mu)^2) = \sum_{\text{all x}} (x - \mu)^2 f(x)$$

However, we seldom know the population variance in practice, so we will only use the sample variance.

The *Standard Deviation* is the square root of the variance, so the standard deviation is σ.

Let's take some template data to look at what the variance (and standard deviation) measure: Consider the data:

$$-\frac{2}{n}, -\frac{1}{n}, 0, \frac{1}{n}, \frac{2}{n}$$

If n is large, our data is tightly packed together about the mean, 0. If n is small, the data are spread out. The variance of this sample is:

$$\sigma^2 = \frac{1}{5}\left(\frac{4+1+0+1+4}{n^2}\right) = \frac{2}{n^2}$$

so that the standard deviation is:

$$\sigma = \frac{\sqrt{2}}{n}$$

and this is in agreement with our heuristic: If n is large, our data is tightly packed about the mean, and the standard deviation is small. If n is small, our data is loosely distributed about the mean, and the standard deviation is large. Another way to look at the standard deviation is in linear algebra terms: If the data is put into a vector of length m (call it $\boldsymbol{x}$), then the standard deviation is:

$$\sigma = \frac{\|\boldsymbol{x}\|}{\sqrt{m}}$$

4.3.1 Covariance and Correlation Coefficients

If we have two data sets, sometimes we would like to compare them to see how they relate to each other.

Definition: Let $X = \{x_1, \ldots, x_m\}, Y = \{y_1, \ldots, y_m\}$ be two data sets with means μ_x, μ_y respectively. Then the *covariance* of the data sets is given by:

$$\mathrm{Cov}(X,Y) = \sigma_{xy}^2 = \frac{1}{m}\sum_{k=1}^{m}(x_k - \mu_x)(y_k - \mu_y)$$

There are exercises at the end of the chapter that will reinforce the notation and give you some methods for manipulating the covariance. In the meantime, it is easy to remember this formula if you think of the following:

If X and Y have mean zero, and we think of X and Y as vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, then the covariance is just the dot product between the vectors, divided by m:

$$\mathrm{Cov}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{m}\boldsymbol{x}^T\boldsymbol{y}$$

We can then interpret what it means for X, Y to have a covariance of zero: $\boldsymbol{x}$ is orthogonal to $\boldsymbol{y}$. Continuing with this analogy, if we normalized by the size of $\boldsymbol{x}$ and the size of $\boldsymbol{y}$, we'd get the cosine of the angle between them. This is the definition of the correlation coefficient, and gives the relationship between the covariance and correlation coefficient:

Definition: The *correlation coefficient* between x and y is given by:

$$\rho_{xy} = \frac{\sigma_{xy}^2}{\sigma_x \sigma_y} = \frac{\sum_{k=1}^{m}(x_k - \mu_x)(y_k - \mu_y)}{\sqrt{\sum_{k=1}^{m}(x_k - \mu_x)^2 \cdot \sum_{k=1}^{m}(y_k - \mu_y)^2}}$$

Again, thinking of X, Y as having zero mean and placing the data in vectors $\boldsymbol{x}, \boldsymbol{y}$, then this formula becomes:

$$\rho_{xy} = \frac{\boldsymbol{x}^T \boldsymbol{y}}{\|\boldsymbol{x}\| \cdot \|\boldsymbol{y}\|} = \cos(\theta)$$

This works out so nicely because we have a $\frac{1}{m}$ in both the numerator and denominator, so they cancel each other out.

We also see immediately that ρ_{xy} can only take on the real numbers between -1 and 1. Some interesting values of ρ_{xy}:

If ρ_{xy} is:	Then the data is:
1	Perfectly correlated ($\theta = 0$)
0	Uncorrelated ($\theta = \frac{\pi}{2}$)
-1	Perfectly (negatively) correlated ($\theta = \pi$)

One last comment before we leave this section: The Covariance and Correlation Coefficient only look for *linear* relationships between data sets!

For example, we know that $\sin(x)$ and $\cos(x)$ (as functions, or as data points sampled at equally spaced intervals) will be uncorrelated, but, because $\sin^2(x) + \cos^2(x) = 1$, we see that $\sin^2(x)$ and $\cos^2(x)$ are perfectly correlated.

This difference is the difference between the words "correlated" and "statistically independent". Statistical independence (not defined here) and correlations are not the same thing! We will look at this difference closely in a later section.

4.4 The Covariance Matrix

If we have p data points in $\mathbb{R}^n$, we can think of the data as a $p \times n$ matrix. Let X denote the *mean-subtracted* data matrix (as we defined previously). A natural question to ask is then how the i^{th} and j^{th} dimensions (columns) covary- so we'll compute the covariance between the i, j columns to define:

$$\sigma_{ij}^2 = \frac{1}{p}\sum_{k=1}^{p} X(k,i) \cdot X(k,j)$$

Computing this for all i, j will result in an $n \times n$ symmetric matrix, C, for which:

$$C_{ij} = \sigma_{ij}^2$$

In the exercises, we have you show that we can conclude that C can be computed using the definition below:

Definition: Let X denote a matrix of data, so that, if X is $p \times n$, then we have p data points in $\mathbb{R}^n$. Furthermore, we assume that the data in X has been mean subtracted. Then the *covariance matrix* associated with X is given by:

$$C = \frac{1}{p} X^T X$$

In Matlab, it is easy to compute the covariance matrix. For your convenience, we repeat the mean-subtraction routine here:

```
%X is a pxn matrix of data:
[p,n]=size(X);
m = mean(X);
Xm = X-repmat(m,p,1);
C=(1/p)*X'*X;
```

Matlab also has a built-in covariance function. It will automatically do the mean-subtraction (which is a lot of extra work if you've already done it!).

```
C=cov(X);
```

If you forget which sizes Matlab uses, you might want to just compute the covariance yourself. It assumes, as we did, that the matrix is $p \times n$, and returns an $n \times n$ covariance- HOWEVER, it will divide by $p-1$ rather than by p. This is not a big issue for the applications we will be considering- you may use either method, but be aware that there are differences in the actual algorithms.

4.5 Exercises

1. By hand, compute the mean and variance of the following set of data:

   $$1, 2, 9, 6, 3, 4, 3, 8, 4, 2$$

2. Obtain a sampling of 1000 points using the uniform distribution: and 1000 points using the normal distribution:

   ```
   x=rand(1000,1);
   y=randn(1000,1);
   ```

   Compare the distributions using Matlab's *hist* command: `hist([x y],100)` and print the results. You'll note that the histograms have not been scaled so that the areas sum to 1, but we do get an indication of the nature of the data.

3. Compute the value of K in the double Laplacian function so that f is a p.d.f.

4. Next, load a sample of human voice: `load laughter` If you type `whos`, you'll see that you have a vector y with the sound data. The computers in the lab do have sound cards, but they don't work very well with Matlab, so we won't listen to the sample. Before continuing, you might be curious about what the data in y looks like, so feel free to plot it. We want to look at the distribution of the data in the vector y, and compare it to the normal distribution. The mean of y is already approximately zero, but to get a good comparison, we'll take a normal distribution with the same variance:

```
clear
load laughter
whos
sound(y,Fs);  %This only works if there's a good sound card
s=std(y);
x=s*randn(size(y));
hist([x y],100);  %Blue is "normal", Red is Voice
```

   Print the result. Note that the normal distribution is much flatter than the distribution of the voice signal.

5. Compute the covariance between the following data sets:

$$\begin{array}{c|ccccccc} x & -1.0 & -0.7 & -0.4 & -0.1 & 0.2 & 0.5 & 0.8 \\ \hline y & -1.3 & -0.7 & -0.1 & 0.5 & 1.1 & 1.7 & 2.3 \end{array} \tag{4.1}$$

6. Let $\mathbf{x}$ be a vector of data with mean μ, and let a, b be scalars. What is the mean of $a\mathbf{x}$? What is the mean of $\mathbf{x} + b$? What is the mean of $a\mathbf{x} + b$? (NOTE: Formally, the addition of a vector and a scalar is not defined. Here, we are utilizing Matlab notation: The result of a vector plus a scalar is addition done componentwise. This is only done with scalars- for example, a matrix added to a vector is still not defined, while it is valid to add a matrix and a scalar).

7. Let $\mathbf{x}$ be a vector of data with variance σ^2, and let a, b be scalars. What is the variance of $a\mathbf{x}$? What is the variance of $\mathbf{x} + b$? What is the variance of $a\mathbf{x} + b$?

8. Show that, for data in vectors $\mathbf{x}$, $\mathbf{y}$ and a real scalar a,

$$\mathrm{Cov}(ax, y) = a\mathrm{Cov}(x, y) \quad \mathrm{Cov}(x, by) = b\mathrm{Cov}(x, y)$$

9. Show that, for data in $\mathbf{x}$ and a vector consisting only of the scalar a,

$$\mathrm{Cov}(x, a) = 0$$

10. Show that, for a and b fixed scalars, and data in vectors $\mathbf{x}, \mathbf{y}$,

$$\mathrm{Cov}(x + a, y + b) = \mathrm{Cov}(x, y)$$

11. If the data sets X and Y are the same, what is the covariance? What is the correlation coefficient? What if $Y = mX$? What if $Y = mX + b$?

12. Let X be a $p \times n$ matrix of data, where we n columns of p data points (you may assume each column has zero mean). Show that the $(i,j)^{\text{th}}$ entry of $\frac{1}{p}X^TX$ is the covariance between the i^{th} and j^{th} columns of X. HINT: It might be convenient to write X in terms of its columns,

$$X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$$

Also show that $\frac{1}{p}X^TX$ is a symmetric matrix.

13. This exercise shows us that our geometric insight might not extend to high dimensional space. We examine how points are distributed in high dimensional hypercubes and unit balls. Before we begin, let us agree that a hypercube of dimension n has the edges:

$$(\pm 1, \pm 1, \pm 1, \ldots, \pm 1)^T$$

so, for example, a 2-d hypercube (a square) has edges:

$$(1,1)^T, (-1,1)^T, (1,-1)^T, (-1,-1)^T$$

(a) Show that the distance (standard Euclidean) from the origin to a corner of a hypercube of dimension d is $\sqrt{d}$. What does this imply about the shape of the "cube", as $d \to \infty$?

(b) The volume of a $d-$dimensional hypersphere of radius a can be written as:

$$V_d = \frac{S_d a^d}{d}$$

where S_d is the $d-$dimensional surface area of the unit sphere.

First, compute the volume between hyperspheres of radius a and radius $a - \epsilon$.

Next, show that the ratio of this volume to the full volume is given by:

$$1 - \left(1 - \frac{\epsilon}{a}\right)^d$$

What happens as $d \to \infty$?

If we have 100,000 data points "uniformly distributed" in a hypersphere of dimension 10,000, where are "most" of the points?

4.6 Linear Regression

In this section, we examine the simplest case of fitting data to a function. We are given two sets of data-

$$X = \{x_1, x_2, \ldots, x_n\} \quad Y = \{y_1, y_2, \ldots, y_n\}$$

We wish to find the best linear relationship between X and Y. But what is "best"? It depends on how you look at the data, as described in the next three exercises.

1. **Exercise:** Let y be a function of x. Then we are trying to find m and b so that
$$y = mx + b$$
best describes the data. If the data were perfectly linear, then this would mean that:
$$\begin{aligned} y_1 &= mx_1 + b \\ y_2 &= mx_2 + b \\ &\vdots \\ y_n &= mx_n + b \end{aligned}$$
However, most of the time the data is not actually, *exactly* linear, so that the values of y don't match the line: $mx + b$. Thus we have an error:
$$E_1 = \sum_{k=1}^{n} |y_k - (mx_k + b)|$$
   (a) Show graphically what this error would represent for one of the data points.
   (b) E_1 is a function of two variables, m and b. What is the difficulty in determining the minimum[1] error using this error function?
   (c) Define the squared error as:
$$E_{\text{se}} = \sum_{k=1}^{n} (y_k - (mx_k + b))^2$$
   Why is it appropriate to use this error instead of the other error?
   (d) E_{se} is a function of m and b, so the minimum value occurs where
$$\frac{\partial E_{\text{se}}}{\partial m} = 0 \quad \frac{\partial E_{\text{se}}}{\partial b} = 0$$
   Show that this leads to the system of equations: (the summation index is 1 to n)
$$\begin{aligned} m \sum x_k^2 + b \sum x_k &= \sum x_k y_k \\ m \sum x_k + bn &= \sum y_k \end{aligned}$$

[1] We could solve this problem by using Linear Programming, but that is outside the scope of this text.

(e) **Exercise:** Write a Matlab routine that will take a $2 \times n$ matrix of data, and output the values of m and b found above. The first line of code should be:

```
function [m,b]=Line1(X)
```

and save as `Line1.m`.

2. **Exercise:** If we treat x as a function of y (i.e., $x = f(y)$), how does that change the equations above? Draw a picture of what the error would represent in this case. Write a new Matlab routine `[m,b]=Line2(X)` to reflect these changes.

3. **Exercise:** The last case is where we treat x and y independently, so that we don't assume that one is a function of the other.

   (a) Show that, if $ax + by + c = 0$ is the equation of the line, then the distance from (x_1, y_1) to the line is

   $$\frac{|ax_1 + by_1 + c|}{\sqrt{a^2 + b^2}}$$

   which is the size of the orthogonal projection of the point to the line. This is actually problem 53, section 11.3 of Stewart's Calculus text, if you'd like more information.
   (HINT: The vector $[a, b]^T$ is orthogonal to the line $ax + by + c = 0$. Take an arbitrary point P on the line, and project an appropriate vector to $[a, b]^T$.)
   Conclude that the error function is:

   $$E = \sum_{k=1}^{n} \frac{(ax_k + by_k + c)^2}{a^2 + b^2}$$

   (b) Draw a picture of the error in this case, and compare it graphically to the error in the previous 2 exercises.

   (c) The optimimum value of E occurs where $\frac{\partial E}{\partial c} = 0$. Show that if we mean subtract X and Y, then we can take $c = 0$. This leaves only two variables.

   (d) Now our error function is:

   $$E = \sum_{k=1}^{n} \frac{(ax_k + by_k)^2}{a^2 + b^2}$$

   Show that we can transform this function (with appropriate assumptions) to:

   $$E = \sum_{k=1}^{n} \frac{(x_k + \mu y_k)^2}{1 + \mu^2}$$

   (for some μ), and conclude that E is a function of one variable.

(e) Now the minimum occurs where $\frac{dE}{d\mu} = 0$. Compute this quantity to get:

$$\mu^2 A + \mu B + C = 0$$

where A, B, C are expressions in $\sum x_k y_k$, $\sum x_k^2$, $\sum y_k^2$. This is a quadratic expression in μ, which we can solve. Why are there (possibly) 2 real solutions?

(f) Write a Matlab routine `[a,b,c]=Line3(X)` that will input a $2 \times n$ matrix, and output the right values of a, b, and c.

4. **Exercise:** Try the 3 different approaches on the following data set, which represents heights (in inches) and weight (in lbs.) of 10 teenage boys. (Available in `HgtWgt.mat`)

X	69	65	71	73	68	63	70	67	69	70
Y	138	127	178	185	141	122	158	135	145	162

Plot the data with the 3 lines. What do the 3 approaches predict for the weight of someone that is 72 inches tall?

5. **Exercise:** Do the same as the last exercise, but now add the data point $(62, 250)$. Compare the new lines with the old. Did things change much?

6. **Matlab Note:** Consider using the command `subplot` to plot multiple graphs on the same figure. For example, try the following sequence of commands:

```
x=linspace(-8,8);
y1=sin(x);
y2=sin(2*x);
y3=sin(x.*x);
y4=sin(exp(-x));
subplot(2,2,1);
plot(x,y1);
subplot(2,2,2);
plot(x,y2);
subplot(2,2,3);
plot(x,y3);
subplot(2,2,4);
plot(x,y4);
```

4.7 The Median-Median Line:

The median of data is sometimes preferable to the mean, especially if there exists a few data points that are far different than "most" data.

1. **Definition:** The *median* of a data set is the value so that exactly half of the data is above that point, and half is below. If you have an odd number of points, the median is the "middle" point. If you have an even number, the median is the average of the two "middle" points. Matlab uses the `median` command.

2. **Exercise:** Compute (by hand, then check with Matlab) the medians of the following data:

   - $$1, 3, 5, 7, 9$$
   - $$1, 2, 4, 9, 8, 1$$

   The motivation for the median-median line is to have a procedure for line fitting that is not as sensitive to "outliers" as the 3 methods in the previous section.

Median-Median Line Algorithm

- Separate the data into 3 equal groups (or as equal as possible). Use the $x-$axis to sort the data.
- Compute the median of each group (first, middle, last).
- Compute the equation of the line through the first and last median points.
- Find the vertical distance between the middle median point and the line.
- Slide the line 1/3 of the distance to the middle median point.

3. **Exercise:** Understand the commands in the program `mmline`. There are several new matlab functions used. Below are some hints as to how we'll divide the data into three groups.

   - If we divide a number by three, we have three possible remainders: $0, 1, 2$. What is the most natural way of seperating data in these three cases (i.e., if we had 27, 28 or 29 data points)?
   - Look at the Matlab command `rem`. Notice that:

     ```
     rem(27,3)=0   rem(28,3)=1   rem(29,3)=2
     ```

   - Look at the Matlab command `sort`. The full command looks like: `[s,index]=sort(x)` The output vector s will be x sorted. The vector `index` will contain which order the original indices were in. That is,

     ```
     x(index)=s
     ```

- We can therefore sort x first, then sort y according to the index for x.

4. **Exercise:** Try this algorithm on the last data set, then add the new data point. Did your lines change as much?

5. **Exercise:** Consider the following data set [**?**] which relates the index of exposure to radioactive contamination from Hanford to the number of cancer deaths per $100,000$ residents. We would like to get a relationship between these data. Use the four techniques above, and compare your answers. Compute the actual errors for the first three types of fits and compare the numbers.

County/City	Index	Deaths
Umatilla	2.5	147
Morrow	2.6	130
Gilliam	3.4	130
Sherman	1.3	114
Wasco	1.6	138
Hood River	3.8	162
Portland	11.6	208
Columbia	6.4	178
Clatsop	8.3	210

Chapter 5

Linear Algebra

The first step in processing data will involve using invertible linear transformations- since they are invertible and relatively easy, we can always reconstruct the original data set. Furthermore, we'll see how the matrix factorization formulas from linear algebra can be used to our advantage in solving "real world" problems.

It can be argued that all of linear algebra can be understood using the *Four Fundamental Subspaces* associated with a matrix. Because these ideas are the foundation, we need an explicit method for analyzing these subspaces- That method will be the *Singular Value Decomposition.* It is unfortunate that most first courses in linear algebra do not cover this material, so we do it here. Again, we cannot stress the importance of this decomposition enough- We will apply this technique throughout the rest of this text.

5.1 Representation, Basis and Dimension

In Chapter 2, we looked at representing data points using different sets of bases. We saw that, if $\mathcal{B} = \{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(k)}\}$ is a basis for the data, then every data point in our set can be written as:

$$\boldsymbol{x}^{(j)} = \sum_{i=1}^{k} \alpha_i^{(j)} \boldsymbol{v}^{(i)}$$

so that every data point in our subset of $\mathbb{R}^n$ is identified with a point in $\mathbb{R}^k$ via:

$$\boldsymbol{x}^{(j)} = (x_1^{(j)}, \ldots, x_n^{(j)}) \longrightarrow (\alpha_1^{(j)}, \ldots, \alpha_j^{(j)}) \doteq [\boldsymbol{x}^{(j)}]_{\mathcal{B}}$$

where we call $[\boldsymbol{x}]_{\mathcal{B}}$ the *coordinates of* $\boldsymbol{x}$ *with respect to the basis* $\mathcal{B}$. Furthermore, if this is a basis for the data, then it is *isomorphic* to $\mathbb{R}^k$. We'll recall the definition:

Definition: Any one-to-one (and onto) linear map is called an isomorphism. In particular, any change of coordinates is an isomorphism. Spaces that are isomorphic have essentially the same algebraic structure- adding vectors in one

space is corresponds to adding vectors in the second space, and scalar multiplication in one space is the same as scalar multiplication in the second.

Using this idea, we can define the **dimension of a subspace**: The dimension of a subspace is the number of basis vectors it requires to represent it. This implies that no matter what basis we choose, it always requires the same number (otherwise, the dimension would depend on the basis, and that wouldn't be good!).

Generically, given a linearly independent spanning set $\mathcal{B}$, to compute the coordinates of a data point with respect to $\mathcal{B}$ requires a matrix inversion (or more generally, Gaussian elimination) to solve the equation:

$$\boldsymbol{x} = \alpha_1 \boldsymbol{v}_1 + \alpha_2 \boldsymbol{v}_2 + \ldots + \alpha_n \boldsymbol{v}_n$$

for the coordinates α_i. We can write this in matrix form as:

$$\boldsymbol{x} = [\boldsymbol{v}_1 \ \ldots \ \boldsymbol{v}_n][\boldsymbol{x}]_{\mathcal{B}} = V[\boldsymbol{x}]_{\mathcal{B}}$$

so that, in the general case where we have n basis vectors of $\mathbb{R}^n$,

$$[\boldsymbol{x}]_{\mathcal{B}} = V^{-1}\boldsymbol{x}$$

If we have fewer than n basis vectors, V will not be square, and thus not invertible. However, if $\boldsymbol{x}$ is contained in the span of the basis, then we will be able to solve for the coordinates of $\boldsymbol{x}$

It gets easier if the basis is orthonormal- before continuing, we'll want to recall the following:

Definition: A real $n \times n$ matrix $\mathbb{Q}$ is said to be *orthogonal* if

$$\mathbb{Q}^T\mathbb{Q} = I$$

This is the property that makes an orthonormal basis nice to work with- it's inverse is its transpose.

In an orthonormal basis, it is easy to compute the coordinates of a vector $\boldsymbol{x}$ with respect to this basis. Note that, if

$$\boldsymbol{x} = \alpha_1 \boldsymbol{u}_1 + \ldots + \alpha_k \boldsymbol{u}_k$$

Then the coordinate α_j is just a dot product:

$$\boldsymbol{x} \cdot \boldsymbol{u}_j = 0 + \ldots + 0 + \alpha_j \boldsymbol{u}_j \cdot \boldsymbol{u}_j + 0 + \ldots 0$$

We can also interpret each individual coordinate as the projection of $\boldsymbol{x}$ onto the appropriate basis vector:

$$\alpha_j = \text{Proj}_{\boldsymbol{u}_j}(\boldsymbol{x}) = \boldsymbol{u}_j \cdot \boldsymbol{x}$$

Writing this in matrix form, let $\boldsymbol{c} = [\alpha_1, \ldots, \alpha_k]^T$ and $U = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k]$, then

$$\boldsymbol{c} = U^T\boldsymbol{x}$$

We summarize our discussion with the following important, but easily proved theorem (the proof is simply to multiply out the expressions and use the properties of an orthonormal basis).

Change of Basis Theorem: Let $\{\boldsymbol{u}_i\}_{i=1}^k$ be a new orthonormal basis for the subspace containing our data (so that this is a subspace of $\mathbb{R}^n$), and let U be the $n \times k$ matrix whose columns are formed from the basis vectors. Then

- *The Coordinates of $\boldsymbol{x}$ with respect to U:*

$$[\boldsymbol{x}]_U = U^T \boldsymbol{x}$$

  which says that a change of coordinates can be performed via matrix multiplication.

- *The Reconstruction of $\boldsymbol{x}$ in $\mathbb{R}^n$ is given by:*

$$\hat{\boldsymbol{x}} = UU^T \boldsymbol{x}$$

  where, if the subspace formed by U contains $\boldsymbol{x}$, then $\boldsymbol{x} = \hat{\boldsymbol{x}}$.

Example Let $\boldsymbol{x} = [3,2,3]^T$ and let the basis vectors be $\boldsymbol{u}_1 = \frac{1}{\sqrt{2}}[1,0,1]^T$ and let $\boldsymbol{u}_2 = [0,1,0]^T$. Then

$$[\boldsymbol{x}]_U = U^T \boldsymbol{x} = \begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 3\sqrt{2} \\ 2 \end{bmatrix}$$

which is the two dimensional representation of the (three dimensional) vector $\boldsymbol{x}$. To get $\boldsymbol{x}$ back into $\mathbb{R}^3$, we simply multiply the coordinates by U. (You should verify this).

5.2 The Four Fundamental Subspaces

Given any $m \times n$ matrix A, we consider the mapping $A : \mathbb{R}^n \to \mathbb{R}^m$ by:

$$\boldsymbol{x} \to A\boldsymbol{x}$$

This concept creates four fundamental subspaces:

Symbol	Definition	Name	Where
$\mathcal{R}(A)$	$\{\boldsymbol{y} \in \mathbb{R}^m \mid \boldsymbol{y} = A\boldsymbol{x}, \boldsymbol{x} \in \mathbb{R}^n\}$	Columnspace of A	$\mathbb{R}^m$
$\mathcal{N}(A)$	$\{\boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} = \boldsymbol{0}\}$	Nullspace of A	$\mathbb{R}^n$
$\mathcal{R}(A^T)$	$\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} = A^T\boldsymbol{y}, \boldsymbol{y} \in \mathbb{R}^m\}$	Row space of A	$\mathbb{R}^n$
$\mathcal{N}(A^T)$	$\{\boldsymbol{y} \in \mathbb{R}^m \mid A^T\boldsymbol{y} = \boldsymbol{0}\}$	Nullspace of A^T	$\mathbb{R}^m$

The column space is also the image of the function- This is clear if we write: $A = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n]$ so that $A\boldsymbol{x} = x_1\boldsymbol{a}_1 + x_2\boldsymbol{a}_2 + \cdots + x_n\boldsymbol{a}_n$ and we see that Ax is a linear combination of the columns of A.

The fundamental subspaces subdivide the domain and range of the mapping in a particularly nice way:

Theorem: Let A be an $m \times n$ matrix. Then:

$$\begin{aligned} \mathcal{N}(A) &\perp \mathcal{R}(A^T) \\ \mathcal{N}(A^T) &\perp \mathcal{R}(A) \end{aligned}$$

Proof: See the exercises.

Definition: The *rank* of a matrix A is the number of independent columns of A..

Rank Theorem: Let the $m \times n$ matrix A have rank r. Then

$$r + \dim(\mathcal{N}(A)) = n$$

This is clear if we consider the matrix under Gaussian elimination. This theorem says that the number of pivot columns plus the other columns (which correspond to free variables) is equal to the total number of columns.

We also have that the dimension of the columnspace is equal to the rank. Now let's do a counting argument to get the dimensions of the rest of the spaces. If the dimension of the columnspace (the rank) is r, then the dimension of the nullspace of A is $n - r$ by the rank theorem. Furthermore, the dimension of the nullspace of A^T must be $m - r$, since there are m dimensions total in the range. Lastly, we also get that the dimension of the columnspace of A^T (or the rowspace) must also be equal to the rank, r. Thus we have shown that the dimension of the columnspace of A must be equal to the dimension of the rowspace. Furthermore, we have half of the theorem below:

Theorem: (We'll wait to show this until after Singular Value Decomposition):

$$\text{rank}(A) = \text{rank}(A^T) = \text{rank}(A^T A) = \text{rank}(AA^T)$$

We might ask what the implications are to a matrix of data- Let's consider some.

Suppose you are given a matrix of data, A, which is $m \times n$. With no other information, we do not know whether we should consider this matrix as n points in $\mathbb{R}^m$, or m points in $\mathbb{R}^n$. In one sense, it doesn't matter! The theorems we've discussed shows that the dimension of the columnspace is equal to the dimension of the rowspace. Later on, we'll find out that if we can find a basis for the columnspace, it is easy to find a basis for the rowspace. We'll need some more machinery first.

5.3 Special Mappings: The Projectors

It will be of benefit to understand certain theorems and algorithms as being projections. We have already seen what an orthogonal projection of one vector to another looks like. We present now a special matrix that performs projections.

Definition: A *Projector* is a square matrix $\mathbb{P}$ so that:

$$\mathbb{P}^2 = \mathbb{P}$$

Example: The following are two projectors. Their matrix representations are given by:

$$P_1 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \qquad P_2 = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Some samples of the projections are given in Figure 5.1, where we see that both project to the subspace spanned by $(1,1)^T$.

Let's consider the action of these matrices on an arbitrary point:

$$P_1\boldsymbol{x} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ x \end{bmatrix}, P_1(P_1\boldsymbol{x}) = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ x \end{bmatrix} = \begin{bmatrix} x \\ x \end{bmatrix}$$

$$P_2\boldsymbol{x} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{x+y}{2} \\ \frac{x+y}{2} \end{bmatrix} = \frac{x+y}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

You should verify that $P_2^2\boldsymbol{x} = P_2(P_2(\boldsymbol{x})) = \boldsymbol{x}$.

You can deduce along which direction a point is projected by drawing a straight line from the point $\boldsymbol{x}$ to the point $\mathbb{P}\boldsymbol{x}$. In general, this direction will depend on the point. We denote this direction by the vector $\mathbb{P}\boldsymbol{x} - \boldsymbol{x}$.

From the previous examples, we see that $\mathbb{P}\boldsymbol{x} - \boldsymbol{x}$ is given by:

$$P_1\boldsymbol{x} - \boldsymbol{x} = \begin{bmatrix} 0 \\ x - y \end{bmatrix}, \text{ and } P_2\boldsymbol{x} - \boldsymbol{x} = \begin{bmatrix} \frac{-x+y}{2} \\ \frac{x-y}{2} \end{bmatrix} = \frac{x-y}{2}\begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

You'll notice that in the case of P_2, $P_2\boldsymbol{x} - \boldsymbol{x} = (P_2 - I)\boldsymbol{x}$ is orthogonal to $P_2\boldsymbol{x}$.

Definition: $\mathbb{P}$ is said to be an *orthogonal* projector if it is a projector, and the range of $\mathbb{P}$ is orthogonal to the range of $(I - \mathbb{P})$. We can show orthogonality by taking an arbitrary point in the range, $\mathbb{P}\boldsymbol{x}$ and an arbitrary point in $(I - \mathbb{P})$, $(I - \mathbb{P})\boldsymbol{y}$, and show the dot product is 0.

There is a property of real projectors that make them nice to work with: They are also symmetric matrices:

Theorem: The (real) projector $\mathbb{P}$ is an orthogonal projector iff $\mathbb{P} = \mathbb{P}^T$. For a proof, see for example, citeLefethen.

Caution: An orthogonal projector need not be an orthogonal matrix. Notice that the projector P_2 from Figure 5.1 was not an orthogonal matrix (that is, $P_2P_2^T \neq I$).

We have two primary sources for projectors:

Example: Let $\boldsymbol{a}$ be an arbitrary, real, non-zero vector. Show that

$$\mathbb{P}\boldsymbol{a} = \frac{\boldsymbol{a}\boldsymbol{a}^T}{\|\boldsymbol{a}\|^2}$$

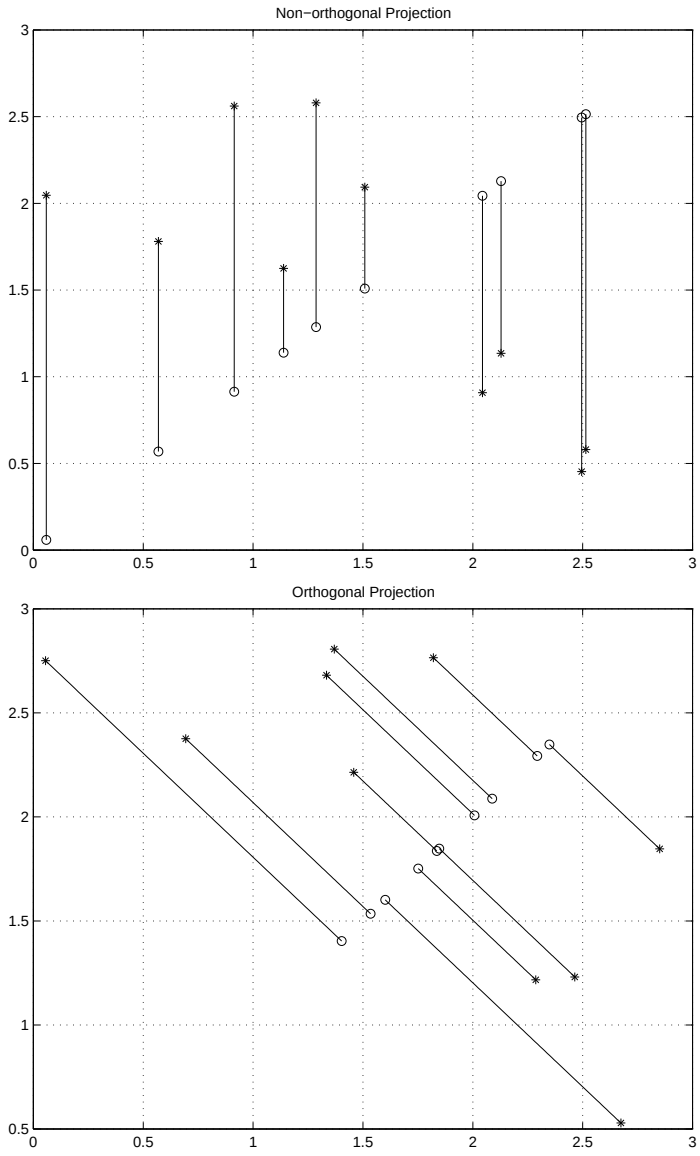


Figure 5.1: Projections P_1 (top picture) and P_2. Asterisks denote the original data point, and circles represent their destination. The line segment follows the direction $P\boldsymbol{x} - \boldsymbol{x}$. Note that P_1 does not project in an orthogonal fashion.

is a rank one orthogonal projector onto the span of $\boldsymbol{a}$.

The matrix $\boldsymbol{a}\boldsymbol{a}^T$ has rank one, since every column is a multiple of $\boldsymbol{a}$.

The matrix is a projector:

$$\mathbb{P}^2 = \frac{\boldsymbol{a}\boldsymbol{a}^T}{\|\boldsymbol{a}\|^2} \cdot \frac{\boldsymbol{a}\boldsymbol{a}^T}{\|\boldsymbol{a}\|^2} = \frac{1}{\|\boldsymbol{a}\|^4}\boldsymbol{a}(\boldsymbol{a}^T\boldsymbol{a})\boldsymbol{a}^T = \frac{\boldsymbol{a}\boldsymbol{a}^T}{\|\boldsymbol{a}\|^2} = \mathbb{P}$$

The matrix is an orthogonal projector, since $\mathbb{P}^T = \mathbb{P}$.

Example: Let $Q = [\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k]$ be a matrix with orthonormal columns. Then

$$\mathbb{P} = QQ^T$$

is an orthogonal projector to the column space of Q. This generalizes the result of the previous exercise. Note that if Q was additionally a square matrix, $QQ^T = I$.

Note that this is exactly the property that we discussed earlier with $U = Q$, where the columns of U formed an orthonormal basis. That is, if $\boldsymbol{x}$ is in the columnspace of U, then

$$\boldsymbol{x} = UU^T\boldsymbol{x}$$

but if $\boldsymbol{x}$ is **not** in the columnspace of U:

$$UU^T\boldsymbol{x} \text{ is the projection of } \boldsymbol{x} \text{ into the columnspace of } U$$

The Best Approximation Theorem If W is a subspace of $\mathbb{R}^n$ and $\boldsymbol{x} \in \mathbb{R}^n$, then the point closest to $\boldsymbol{x}$ in W is the orthogonal projection of $\boldsymbol{x}$ into W. We prove this in the exercises below.

5.4 Exercises

1. Show that $\mathcal{N}(A) \perp \mathcal{R}(A^T)$. You must show that, for arbitrary $\boldsymbol{x}_1 \in \mathcal{N}(A)$ and $\boldsymbol{x}_2 \in \mathcal{R}(A^T)$, we have $(\boldsymbol{x}_1, \boldsymbol{x}_2) = 0$. Hint: Write A in terms of its rows.

2. If A is $m \times n$, how big can the rank of A possibly be?

3. Show that multiplication by an orthogonal matrix preserves lengths: $\|\mathbb{Q}\boldsymbol{x}\|_2 = \|\boldsymbol{x}\|_2$ (Hint: Use properties of inner products). Conclude that multiplication by $\mathbb{Q}$ represents a rigid rotation.

4. Prove the Pythagorean Theorem by induction: Given a set of n orthogonal vectors $\{\boldsymbol{x}_i\}$

$$\|\sum_{i=1}^{n} \boldsymbol{x}_i\|_2^2 = \sum_{i=1}^{n} \|\boldsymbol{x}_i\|_2^2$$

5. Let A be an $m \times n$ matrix where $m > n$, and let A have rank n. Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$, such that $\boldsymbol{y}$ is the orthogonal projection of $\boldsymbol{x}$ onto the columnspace of A. We want a formula for the projector $\mathbb{P} : \mathbb{R}^m \to \mathbb{R}^m$ so that $\mathbb{P}\boldsymbol{x} = \boldsymbol{y}$.

   (a) Why is the projector not $\mathbb{P} = AA^T$?

   (b) Since $\boldsymbol{y} - \boldsymbol{x}$ is orthogonal to the range of A, show that

   $$A^T(\boldsymbol{y} - \boldsymbol{x}) = \boldsymbol{0} \tag{5.1}$$

   (c) Show that there exists $\boldsymbol{v}$ so that Equation (5.1) can be written as:

   $$A^T(A\boldsymbol{v} - \boldsymbol{x}) = 0 \tag{5.2}$$

   (d) Argue that A^TA is invertible, so that Equation (5.2) implies that

   $$\boldsymbol{v} = \left(A^TA\right)^{-1} A^T\boldsymbol{x}$$

   (e) Finally, show that this implies that

   $$\mathbb{P} = A\left(A^TA\right)^{-1} A^T$$

   Note: If A has rank $k < m$, then we will need something different, since A^TA will not be full rank. The missing piece is the singular value decomposition, to be discussed later.

6. The Orthogonal Decomposition Theorem: if $\boldsymbol{x} \in \mathbb{R}^n$ and W is a (non-zero) subspace of $\mathbb{R}^n$, then $\boldsymbol{x}$ can be written *uniquely* as

   $$\boldsymbol{x} = \boldsymbol{w} + \boldsymbol{z}$$

   where $\boldsymbol{w} \in W$ and $\boldsymbol{z} \in W^\perp$.

   To prove this, let $\{\boldsymbol{u}_i\}_{i=1}^p$ be an orthonormal basis for W, define $\boldsymbol{w} = (\boldsymbol{x}, \boldsymbol{u}_1)\boldsymbol{u}_1 + \ldots + (\boldsymbol{x}, \boldsymbol{u}_p)\boldsymbol{u}_p$, and define $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{w}$. Then:

   (a) Show that $\boldsymbol{z} \in W^\perp$ by showing that it is orthogonal to every $\boldsymbol{u}_i$.

   (b) To show that the decomposition is unique, suppose it is not. That is, there are two decompositions:

   $$\boldsymbol{x} = \boldsymbol{w}_1 + \boldsymbol{z}_1, \quad \boldsymbol{x} = \boldsymbol{w}_2 + \boldsymbol{z}_2$$

   Show this implies that $\boldsymbol{w}_1 - \boldsymbol{w}_2 = \boldsymbol{z}_2 - \boldsymbol{z}_1$, and that this vector is in both W and $W^\perp$. What can we conclude from this?

7. Use the previous exercises to prove the **The Best Approximation Theorem** If W is a subspace of $\mathbb{R}^n$ and $\boldsymbol{x} \in \mathbb{R}^n$, then the point closest to $\boldsymbol{x}$ in W is the orthogonal projection of $\boldsymbol{x}$ into W.

5.5 The Decomposition Theorems

5.5.1 The Eigenvector/Eigenvalue Decomposition

1. **Definition:** Let A be an $n \times n$ matrix. Then an eigenvector-eigenvalue pair is $\boldsymbol{v} \neq \mathbf{0}, \lambda$ where

$$A\boldsymbol{v} = \lambda\boldsymbol{v} \Rightarrow (A - \lambda I)\boldsymbol{v} = \mathbf{0} \tag{5.3}$$

2. **Remark:** If Equation (5.3) has a nontrivial solution, then

$$\det(A - \lambda I) = 0$$

which leads to solving for the roots of a polynomial of degree n. This polynomial is called the *characteristic* polynomial.

3. **Remark:** We solve for the eigenvalues first, then solve for the nullspace of $(A - \lambda_i I)$ by solving

$$(A - \lambda_i I)\boldsymbol{x} = \mathbf{0}$$

4. **Remark:** Note that it is possible that one eigenvalue is repeated. This may or may not correspond with the same number of eigenvectors.

5. **Definition:** If eigenvalue λ is repeated k times, then the *algebraic multiplicity* of λ is k.

6. **Definition:** If eigenvalue λ has k associated independent eigenvectors, λ has *geometric multiplicity* k.

7. **Theorem:** If a_λ is the algebraic multiplicity of λ and g_λ is the geometric multiplicity, then

$$a_\lambda \geq g_\lambda$$

8. **Definition:** If, for some eigenvalue λ of A, we have that $a_\lambda > g_\lambda$, A is said to be defective.

9. **Definition:** The set of independent eigenvectors associated with an eigenvector λ, together with $\mathbf{0}$ forms a vector space. This space is called the *eigenspace*, and is denoted by E_λ.

10. **Theorem:** If X is square and invertible, then A and $X^{-1}AX$ have the same eigenvalues.

11. **Exercise:** Prove the previous theorem.

12. **Remark:** One method of characterizing eigenvalues in terms of the determinant and trace of a matrix:

$$\det(A) = \Pi_{i=1}^{n}\lambda_i \quad \text{trace}(A) = \sum_{i=1}^{\infty}\lambda_i$$

13. **Remark:** We will be especially interested in symmetric matrices. The rest of this section is devoted to them.

14. **Definition:** A matrix A is *orthogonally diagonalizeable* if there is an orthogonal matrix $\mathbb{Q}$ and diagonal matrix D so that so that $A = \mathbb{Q}D\mathbb{Q}^T$.

15. **The Spectral Theorem:** If A is an $n \times n$ symmetric matrix, then:

    (a) A has n real eigenvalues (counting multiplicity).

    (b) For all λ, $a_\lambda = g_\lambda$.

    (c) The eigenspaces are mutually orthogonal.

    (d) A is orthogonally diagonalizeable, with $D = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$.

    NOTE: We assume that inside each eigenspace, we have an orthonormal basis of eigenvectors. This is not a restriction, since we can always construct such a basis using Gram-Schmidt.

16. **The Spectral Decomposition:** Since A is orthogonally diagonalizeable, then

$$A = (\boldsymbol{q}_1\ \boldsymbol{q}_2\ \ldots\ \boldsymbol{q}_n) \begin{pmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_n \end{pmatrix} \begin{pmatrix} \boldsymbol{q}_1^T \\ \boldsymbol{q}_2^T \\ \vdots \\ \boldsymbol{q}_n^T \end{pmatrix}$$

    so that:

$$A = (\lambda_1\boldsymbol{q}_1\ \lambda_2\boldsymbol{q}_2\ \ldots\ \lambda_n\boldsymbol{q}_n) \begin{pmatrix} \boldsymbol{q}_1^T \\ \boldsymbol{q}_2^T \\ \vdots \\ \boldsymbol{q}_n^T \end{pmatrix}$$

    so finally:

$$A = \lambda_1\boldsymbol{q}_1\boldsymbol{q}_1^T + \lambda_2\boldsymbol{q}_2\boldsymbol{q}_2^T + \ldots + \lambda_n\boldsymbol{q}_n\boldsymbol{q}_n^T$$

    That is, A is a sum of n rank one matrices, each of which is a projection matrix.

17. **Matlab Exercise:** Verify the spectral decomposition for a symmetric matrix. Type the following into Matlab (the lines that begin with a % denote comments that do not have to be typed in).

```
%Construct a random, symmetric, 6 x 6 matrix:
for i=1:6
  for j=1:i
    A(i,j)=rand;
```

```
    A(j,i)=A(i,j);
  end
end

%Compute the eigenvalues of A:
[Q,L]=eig(A);   %NOTE:  A = Q L Q'
                %L is a diagonal matrix

%Now form the spectral sum
S=zeros(6,6); for i=1:6
  S=S+L(i,i)*Q(:,i)*Q(:,i)';
end

max(max(S-A))
```

Note that the maximum of $S - A$ should be a very small number! (By the spectral decomposition theorem).

5.5.2 The Singular Value Decomposition

There is a special matrix factorization that is extremely useful, both in applications and in proving theorems. This is mainly due to two facts, which we shall investigate in this section: (1) We can use this factorization on *any* matrix, (2) The factorization defines explicitly the rank of the matrix, and all four matrix subspaces.

In what follows, assume that A is an $m \times n$ matrix (so A maps $\mathbb{R}^n$ to $\mathbb{R}^m$).

1. **Remark:** Although A itself is not symmetric, A^TA is $n \times n$ and symmetric. Therefore, it is orthogonally diagonalizable. Let $\{\lambda_i\}_{i=1}^n$ and $V = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n]$ be the eigenvalues and orthonormal eigenvectors.

2. **Exercise:** Show that $\lambda_i \geq 0$ for $i = 1..n$ by showing that $\|A\boldsymbol{v}_i\|_2^2 = \lambda_i$.

3. **Definition:** We define the singular values of A by:

$$\sigma_i = \sqrt{\lambda_i}$$

   where λ_i is an eigenvalue of A^TA.

4. **Remark:** In the rest of the section, we will assume any list (or diagonal matrix) of eigenvalues of A^TA (or singular values of A) will be ordered from highest to lowest: $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$.

5. **Exercise:** Prove that, if $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are distinct eigenvectors of A^TA, then their corresponding images, $A\boldsymbol{v}_i$ and $A\boldsymbol{v}_j$, are orthogonal.

6. **Exercise:** Prove that, if $\boldsymbol{x} = \alpha_1\boldsymbol{v}_1 + \ldots \alpha_n\boldsymbol{v}_n$, then

$$\|A\boldsymbol{x}\|^2 = \alpha_1^2\lambda_1 + \ldots + \alpha_n^2\lambda_n$$

7. **Exercise:** Let W be the subspace generated by the basis $\{\boldsymbol{v}_j\}_{j=k+1}^{n}$, where $\boldsymbol{v}_j$ are the eigenvectors associated with the *zero* eigenvalues of A^TA (therefore, we are assuming that the first k eigenvalues are NOT zero). Show that $W = \text{Null}(A)$.

8. **Exercise:** Prove that if the rank of A^TA is r, then so is the rank of A.

9. **Remark:** Define
$$\boldsymbol{u}_i = \frac{1}{\|A\boldsymbol{v}_i\|_2} A\boldsymbol{v}_i = \frac{1}{\sigma_i} A\boldsymbol{v}_i$$
and let U be the matrix whose i^{th} column is $\boldsymbol{u}_i$.

10. **Remark:** This definition only makes sense sense for the first r vectors $\boldsymbol{v}$ (otherwise, $A\boldsymbol{v}_i = \boldsymbol{0}$). Thus, we'll have to extend the basis to span all of $\mathbb{R}^m$.

11. **Exercise:** Sketch how you might do this.

12. **Remark:** So far, we have shown how to construct two matrices, U and V given a matrix A. That is, the matrix V is constructed by the eigenvectors of A^TA, and the matrix U can be constructed using the $\boldsymbol{v}$'s.

13. **Exercise:** Let A be $m \times n$. Define the $m \times n$ matrix
$$\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n)$$
where σ_i is the i^{th} singular value of the matrix A. Show that
$$AV = U\Sigma$$

14. **The Singular Value Decomposition (SVD)** Let A be any $m \times n$ matrix of rank r. Then
$$A = USV^T$$
where U, S, V are the matrices defined in the previous exercises. That is, U is an orthogonal $m \times m$ matrix, S is a diagonal $m \times n$ matrix, and V is an orthogonal $n \times n$ matrix. The $\boldsymbol{u}$'s are called the *left singular vectors* and the $\boldsymbol{v}$'s are called the *right singular vectors.*

15. **Remark:** Keep in mind the following relationship between the right and left singular vectors:
$$\begin{aligned} A\boldsymbol{v}_i &= \sigma_i \boldsymbol{u}_i \\ A^T\boldsymbol{u}_i &= \sigma_i \boldsymbol{v}_i \end{aligned}$$

16. **Computing The Four Subspaces to a matrix** A. Let $A = USV^T$ be the SVD of A which has rank r. Be sure that the singular values are ordered from highest to lowest. Then:

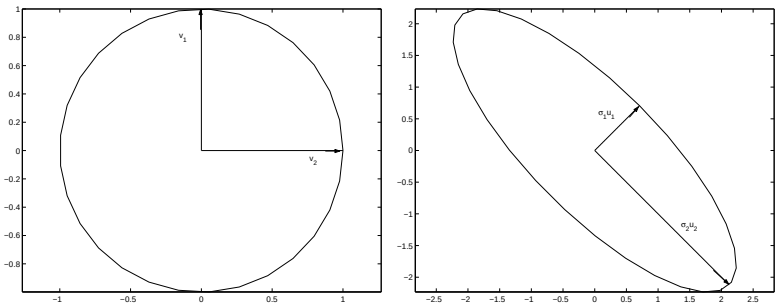


Figure 5.2: The geometric meaning of the right and left singular vectors of the SVD decomposition. Note that $A\boldsymbol{v}_i = \sigma_i \boldsymbol{u}_i$. The mapping $x \to Ax$ will map the unit circle on the left to the ellipse on the right.

(a) A basis for $\mathcal{R}(A)$ is $\{\boldsymbol{u}_i\}_{i=1}^{r}$

(b) A basis for $\mathcal{N}(A)$ is $\{\boldsymbol{v}_i\}_{i=r+1}^{n}$

(c) A basis for $\mathcal{R}(A^T)$ is $\{\boldsymbol{v}_i\}_{i=1}^{r}$

(d) A basis for $\mathcal{N}(A^T)$ is $\{\boldsymbol{u}_i\}_{i=r+1}^{m}$

17. We can also visualize the right and left singular values as in Figure 5.2. We think of the $\boldsymbol{v}_i$ as a special orthogonal basis in R^n (Domain) that maps to the ellipse whose axes are defined by $\sigma_i \boldsymbol{u}_i$.

18. The SVD is one of the premier tools of linear algebra, because it allows us to completely reveal everything we need to know about a matrix mapping: The rank, the basis of the Nullspace, a basis for the column space, the basis for the Nullspace of A^T, and of the row space. This is depicted in Figure 5.3.

19. Lastly, the SVD provides a decomposition of any linear mapping into two rotations and a scaling. This will become important later when we try to deduce a mapping matrix from data (See the section on *signal separation*).

20. **Exercise:** Compute the SVD by hand of the following matrices:

$$\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \qquad \begin{pmatrix} 0 & 2 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

21. **Remark:** If m or n is very large, it might not make sense to keep the full matrix U and V.

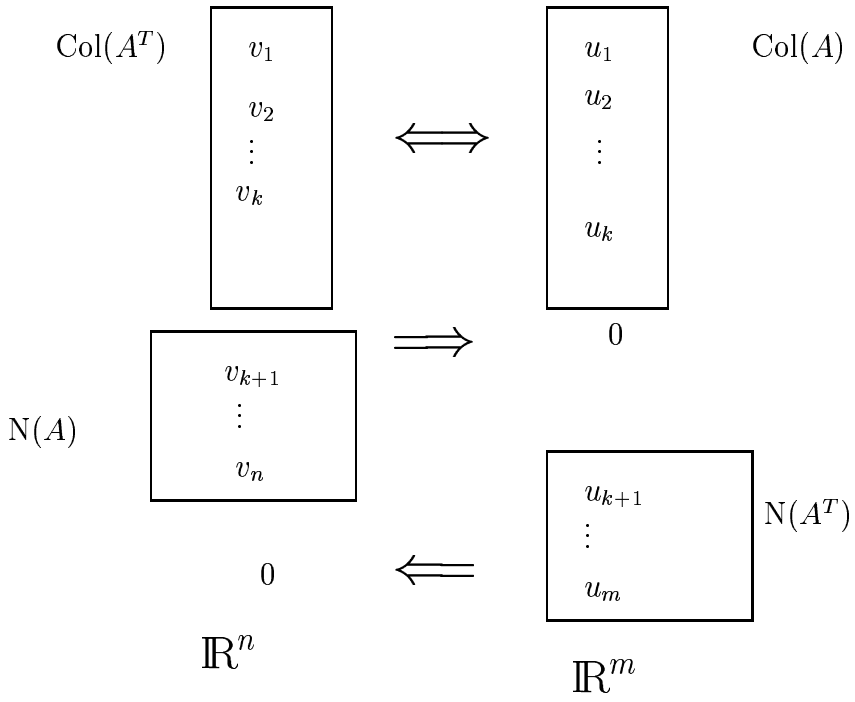


Figure 5.3: The SVD of A (`[U,S,V]=svd(A)`) completely and explicitly describes the 4 fundamental subspaces associated with the matrix, as shown. We have a one to one correspondence between the rowspace and columnspace of A, the remaining $\boldsymbol{v}$'s map to zero, and the remaining $\boldsymbol{u}$'s map to zero (under A^T).

22. **The Reduced SVD** Let A be $m \times n$ with rank r. Then we can write:

$$A = \tilde{U}\tilde{S}\tilde{V}^T$$

where $\tilde{U}$ is an $m \times r$ matrix with orthogonal columns, $\tilde{S}$ is an $r \times r$ *square* matrix, and $\tilde{V}$ is an $n \times r$ matrix.

23. **Theorem:** (Actually, this is just another way to express the SVD). Let $A = USV^T$ be the SVD of A, which has rank r. Then:

$$A = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$$

Therefore, we can approximate A by the sum of rank one matrices.

24. **Matlab and the SVD** Matlab has the SVD built in. The function specifications are: `[U,S,V]=svd(A)` and `[U,S,V]=svd(A,0)` where the first function call returns the full SVD, and the second call returns the reduced SVD.

25. **Matlab Exercise:** Image Processing and the SVD. First, in Matlab, load the clown picture:

```
load clown
```

This loads a matrix X and a colormap, *map*, into the workspace. To see the clown, type:

```
image(X); colormap(map)
```

We now perform a Singular Value Decomposition on the clown. Type in:

```
[U,S,V]=svd(X);
```

How many vectors are needed to retain a good picture of the clown? Try performing a $k-$dimensional reconstruction of the image by typing:

```
H=U(:,1:k)*S(1:k,1:k)*V(:,1:k)'; image(H)
```

for $k = 5, 10, 20$ and 30. What do you see?

Generalized Inverses

Let a matrix A be $m \times n$ with rank r. In the general case, A does not have an inverse. Is there a way of restricting the domain and range of the mapping $\boldsymbol{y} = A\boldsymbol{x}$ so that the map is invertible?

We know that the columnspace and rowspace of A have the same dimensions. Therefore, there exists a 1-1 and onto map between these spaces, and this is our restriction.

To "solve" $\boldsymbol{y} = A\boldsymbol{x}$, we replace $\boldsymbol{y}$ by its orthogonal projection to the columnspace of A, $\hat{\boldsymbol{y}}$. This gives the least squares solution, which makes the problem solvable. To get a unique solution, we replace $\boldsymbol{x}$ by its projection to the rowspace of A, $\hat{\boldsymbol{x}}$. The problem

$$\hat{\boldsymbol{y}} = A\hat{\boldsymbol{x}}$$

now has a solution, and that solution is unique. We can rewrite this problem now in terms of the **reduced SVD** of A:

$$\hat{\boldsymbol{x}} = VV^T\boldsymbol{x}, \qquad \hat{\boldsymbol{y}} = UU^T\boldsymbol{y}$$

Now we can write:

$$UU^T\boldsymbol{y} = U\Sigma V^T\left(VV^T\boldsymbol{x}\right)$$

so that

$$V\Sigma^{-1}U^T\boldsymbol{y} = VV^T\boldsymbol{x}$$

(Exercise: Verify that these computations are correct!)

Given an $m \times n$ matrix A, define its pseudoinverse, $A^\dagger$ by:

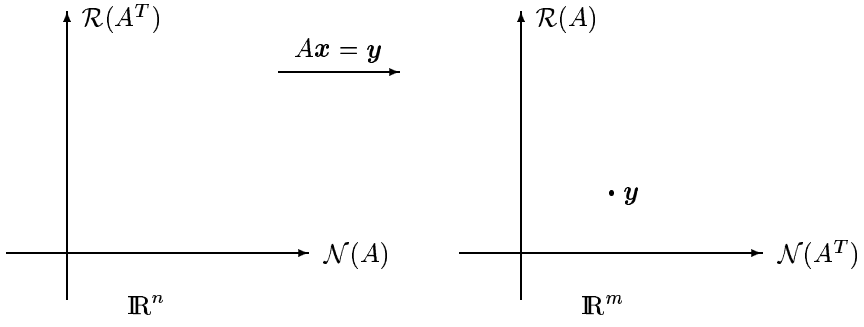
$$A^\dagger = V\Sigma^{-1}U^T$$

We have shown that the least squares solution to $\boldsymbol{y} = A\boldsymbol{x}$ is given by:

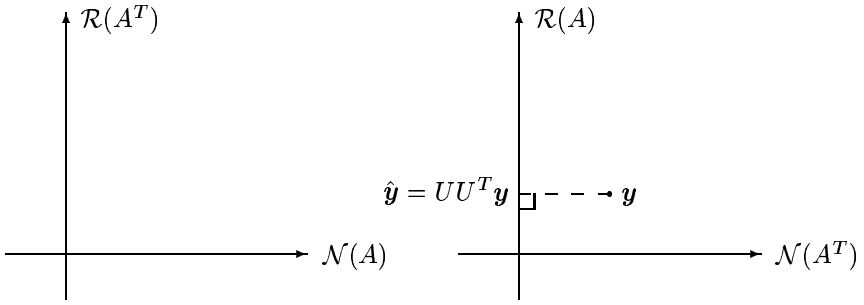
$$\hat{\boldsymbol{x}} = A^\dagger\boldsymbol{y}$$

where $\hat{\boldsymbol{x}}$ is in the rowspace of A, and its image, $A\hat{\boldsymbol{x}}$ is the projection of $\boldsymbol{y}$ into the columnspace of A.

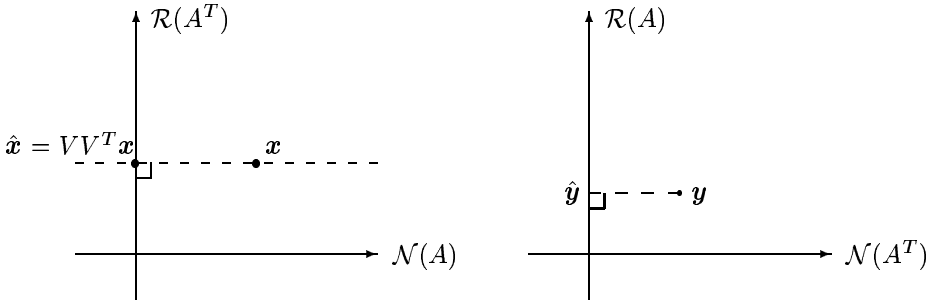
Geometrically, we can understand these computations in terms of the four fundamental subspaces.



In this case, there is no value of $\boldsymbol{x} \in \mathbb{R}^n$ which will map onto $\boldsymbol{y}$, since $\boldsymbol{y}$ is outside the columnspace of A. To get a solution, we project $\boldsymbol{y}$ onto the columnspace of A as shown below:



Now it is possible to find an $\boldsymbol{x}$ that will map onto $\hat{\boldsymbol{y}}$, but if the nullspace of A is nontrivial, then all of the points on the dotted line will also map to $\hat{\boldsymbol{y}}$



Finally, we must choose a unique value of $\boldsymbol{x}$ for the mapping- We choose the $\boldsymbol{x}$ inside the rowspace of A.

This is a very useful idea, and it is one we will explore in more detail later. For now, notice that to get this solution, we analyzed our four fundamental subspaces in terms of the basis vectors given by the SVD.

Exercises

1. Consider

$$\begin{bmatrix} 2 & 1 & -1 \\ 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$$

   (a) Before solving this problem, what are the dimensions of the four fundamental subspaces?

   (b) Use Matlab to compute the SVD of the matrix A, and solve the problem by computing the pseudoinverse of A directly.

   (c) Check your answer explicitly and verify that $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$ are in the rowspace and columnspace. (Hint: If a vector $\boldsymbol{x}$ is already in the rowspace, what is $VV^T\boldsymbol{x}$?)

2. Consider

$$\begin{bmatrix} 2 & 1 & -1 & 3 \\ -1 & 0 & 1 & -2 \\ 7 & 2 & -5 & 12 \\ -3 & -2 & 0 & -4 \\ 4 & 1 & -3 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x4 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ 0 \\ -2 \\ 6 \end{bmatrix}$$

   (a) Find the dimensions of the four fundamental subspaces by using the SVD of A (in Matlab).

   (b) Solve the problem.

   (c) Check your answer explicitly and verify that $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$ are in the rowspace and columnspace.

3. Write the following in Matlab to reproduce Figure 5.2:

```
theta=linspace(0,2*pi,30);
z=exp(i*theta);
X=[real(z);imag(z)]; %The domain points
m=1/sqrt(2);
A=(m*[1,1;1,-1])*[1,0;0,3];
Y=A*X;  %The image of the circle

t=linspace(0,1);
vec1=[0;0]*(1-t)+[0;1]*t;  %The basis vectors v
vec2=[0;0]*(1-t)+[1;0]*t;

Avec1=A*vec1; Avec2=A*vec2; %Image of the basis vectors

figure(1) %The domain
plot(X(1,:),X(2,:),'k',vec1(1,:),vec1(2,:),'k',
     vec2(1,:),vec2(2,:),'k');
axis equal
figure(2) %The image
plot(Y(1,:),Y(2,:),'k',Avec1(1,:),Avec1(2,:),'k',
     Avec2(1,:),Avec2(2,:),'k');
axis equal
```

4. In the previous example, what was the matrix A? The vectors $\boldsymbol{v}$? The vectors $\boldsymbol{u}$? The singular values σ_1, σ_2?

   Once you've written these down from the program, perform the SVD of A in Matlab. Are the vectors the same that you wrote down?

   NOTE: These show that the singular vectors are not unique- they vary by $\pm\boldsymbol{v}$, *or* $\pm\boldsymbol{u}$.